

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Depth on Demand: Streaming Dense Depth from a Low Frame Rate Active Sensor

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Conti, A., Poggi, M., Cambareri, V., Mattoccia, S. (2025). Depth on Demand: Streaming Dense Depth from a Low Frame Rate Active Sensor. Berlin, Heidelberg : Springer-Verlag [10.1007/978-3-031-73030-6_16].

Availability:

This version is available at: <https://hdl.handle.net/11585/1010345> since: 2025-03-26

Published:

DOI: http://doi.org/10.1007/978-3-031-73030-6_16

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

(Article begins on next page)

Depth on Demand: Streaming Dense Depth from a Low Frame Rate Active Sensor

Andrea Conti ¹, Matteo Poggi ¹, Valerio Cambarelli ², and Stefano Mattocchia ¹

¹University of Bologna, Italy

²Sony Depthsensing Solutions, Brussels, Belgium

https://andreaconti.github.io/projects/depth_on_demand

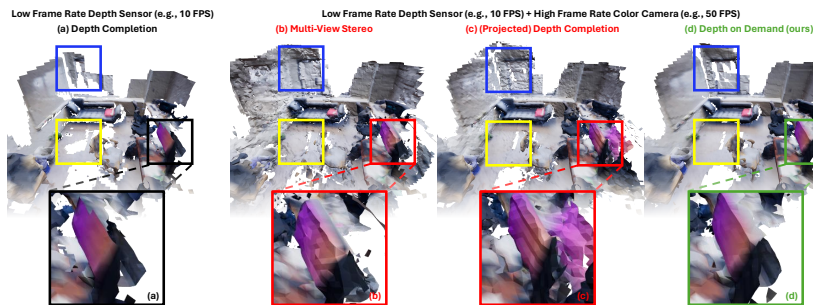


Fig. 1: 3D reconstruction with a low frame rate, sparse depth sensor. Running depth completion (a) on low FPS, sparse depth maps generate holes in the final reconstruction. Adding a higher FPS color camera allows for obtaining depth from Multi-View Stereo (b) or projecting depth to nearby color views and running completion (c), with unsatisfactory results. Our framework (d) performs *temporal* completion using two views and one sparse depth frame, yielding denser and more accurate meshes.

Abstract. High frame rate and accurate depth estimation plays an important role in several tasks crucial to robotics and automotive perception. To date, this can be achieved through ToF and LiDAR devices for indoor and outdoor applications, respectively. However, their applicability is limited by low frame rate, energy consumption, and spatial sparsity. Depth on Demand (DoD) allows for accurate temporal and spatial depth densification achieved by exploiting a high frame rate RGB sensor coupled with a potentially lower frame rate and sparse active depth sensor. Our proposal jointly enables lower energy consumption and denser shape reconstruction, by significantly reducing the streaming requirements on the depth sensor thanks to its three core stages: i) multi-modal encoding, ii) iterative multi-modal integration, and iii) depth decoding. We present extended evidence assessing the effectiveness of DoD on indoor and outdoor video datasets, covering both environment scanning and automotive perception use cases.

1 Introduction

We introduce the intrinsic issues related to active depth sensing and how our framework addresses them widening its applicability to different scenarios.

Active Depth Sensing. In the last decade, RGB-D camera systems have become prominent in fields such as robotics, automotive, and augmented reality, and have scaled down from Kinect v1 to mobile handheld devices such as the Apple iPad. In such systems, one or more conventional RGB cameras are coupled with an *active* depth sensor, *i.e.*, a device that leverages active illumination to infer the 3D structure of the framed scene [41]. Among these sensors, Time-of-Flight (ToF) cameras infer the distance by emitting modulated infrared light into the scene and measuring its return time [1, 3, 41]. On the other hand, Light Detection And Ranging (LiDAR) sensors allow for long-range measurements up to hundreds of meters with or without sunlight at much higher energy consumption and footprint.

Limitations. Despite the reconstruction accuracy of active depth sensors, their inherent structure places limits on their usability. ToF sensors are mainly used for mobile devices and can achieve high frame rates, but imply high energy consumption compared to the limited available battery and overheating. Usually, a drastic reduction of their frame rate is required since it corresponds to a drastic reduction in energy consumption. On the other hand, LiDAR sensors are bulky devices mainly used for autonomous driving, and their moving mechanical components (scanning mirrors) limit the frame rate. Finally, both these technologies manifest spatial sparsity, generating meaningful predictions only for specific spatial locations. Such sparsity can intentionally be induced to minimize acquisition time (in scanned LiDAR [2, 37]) or energy consumption (in ToF [29, 36]). Due to these constraints, the adoption of RGB-D camera systems is difficult in various scenarios. Indeed, Augmented Reality (AR) requires extremely low-power camera systems to fit severely constrained energy budgets. Autonomous driving requires high frame rate depth perception to allow reactive safety-critical applications. 3D shape reconstruction from video streams benefits high frame rate reconstruction to achieve dense meshes without requiring very slow movements from the operator.

Proposal. This paper proposes Depth on Demand (DoD), a framework addressing the three major issues related to active depth sensors in streaming dense depth maps – *i.e.* spatial sparsity, energy consumption, and limited frame rate. The spatial resolution problem represents a well-known issue deeply investigated by the research community through depth completion [26]. On the other hand, energy footprint and low frame rate issues have often been ignored in the literature, although prominent in the deployment of RGB-D systems. We start from the observation that reducing the active sensor temporal resolution – *i.e.* its frame rate – power consumption can be modulated accordingly. Indeed, ToF sensors’ energy consumption scales almost linearly with frame rate [6]. DoD allows coupling together an active depth sensor and an RGB camera to stream dense depth at the RGB camera frame rate, which may be much higher than the former one. The benefit is twofold. On the one hand, it allows the adaptation of

active depth sensor energy consumption to the task specifications, thus meeting the energy constraints of the ToF use case. On the other, it unlocks frame rates higher than the maximum attainable by the active depth sensor itself – this effectively tackles the LiDAR use case, where acquisition is limited to 10 Hz while RGB cameras can easily attain 30 Hz or more. Increasing the depth perception frame rate is of great interest in safety-critical applications such as autonomous driving. However, decoupling frame rates benefits also 3D scene reconstruction as it reduces energy consumption and allows for denser reconstructions. This is showcased in Figure 1: by performing depth completion only at a low frame rate (a) several holes appear in the mesh. Integrating information from a higher frame rate RGB camera (b-d) produces denser meshes. A simple solution to achieve the latter would be relying on Multi-View Stereo algorithms without using depth sensor data (b), or performing depth completion by projecting previous sparse depth points (c). Both these approaches introduce several artifacts as in the highlighted boxes. DoD produces denser and more accurate reconstructions (d). To summarize, our main contributions are as follows:

- We introduce the task of temporal depth stream densification, in which we aim to match the spatial and temporal resolution of a sparse active sensor with one of a higher frame rate RGB camera.
- We design a deep architecture devoted to this purpose, exploiting sparse depth measurements and RGB data collected at time $t - n$ to obtain a dense depth map aligned with the RGB frame at time t .
- We evaluate the proposed framework on several datasets featuring RGB-D video streams and compare it with existing approaches compatible with the outlined setting, proving the superiority of our framework.

2 Related Work

Our proposal intersects both depth perception from a single RGB-D frame and multiple RGB-only views. Little research exists in the literature concerning the integration of the two for spatiotemporal depth perception densification.

Depth Completion. Depth completion aims at densifying a single monocular sparse depth map obtained by an active depth sensor. These methods can be classified into unguided [16, 35, 51] and RGB-guided techniques [27, 32, 49, 55]. To date, the state-of-the-art exploits a single RGB-aligned view to better guide the completion procedure and can be distinguished by the propagation method used. Common taxonomies [26, 41] define early-fusion [15, 28, 45], late-fusion [17, 31, 55], explicit 3D representation [60], residual models [23], and Spatial Propagation Networks (SPN)-based models [8, 9, 13, 32, 38]. SPN-based methods are the most effective and have been extensively studied. Initially, [8, 9] proposed to refine depth through a set of propagating iterations exploiting 3×3 local affinity matrices learned by a UNet. [38] generalized such a framework introducing deformable sampling in the propagation process. [32] introduced an attention-based approach to this sampling, while [54] introduced a novel Geometric SPN module. On a parallel track, [56] developed an unsupervised framework.

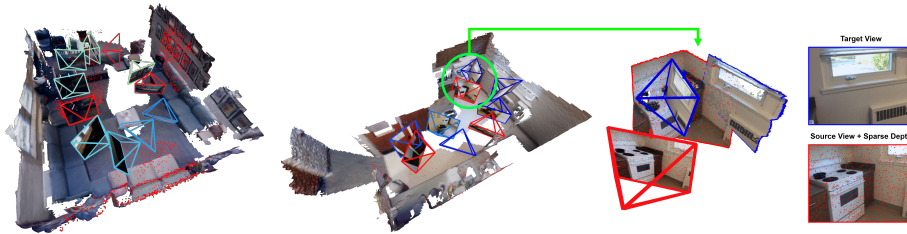


Fig. 2: Temporal Depth Stream Densification Setup. On the left, an example of DoD applied to an indoor video sequence where only a few frames (red views) are associated with sparse depth data. On the right, a close-up example of the supposed setup. Using an RGB-D video stream with only a few sparse depth frames requires the integration of monocular, multi-view, and sparse depth cues. Our framework smoothly enables the recovery of temporal and spatial depth resolution in such a scenario.

DoD greatly differs from depth completion in that it leverages multi-view cue integration. Indeed, depth completion greatly suffers from input depth outliers, since monocular cues alone are insufficient to effectively filter them out. Conversely, multi-view cues enable ignoring wrong sparse depth points which may occur due to depth projection from a previous depth frame. Moreover, state-of-the-art depth completion methods do not usually integrate techniques to deal with such outliers explicitly [34].

Multi-View Stereo. Multi-view depth perception aims at recovering the structure of scenes given overlapping projections of the 3D space on 2D posed images. It can be applied to reconstructing either a 3D model of an object – *e.g.* as a point cloud – or of environments such as indoor scenes. Traditional methods perform such a reconstruction through triangulation and manually engineered features [5, 18, 20, 44]. However, state-of-the-art approaches all exploit learned frameworks. Cost volume-based methods exploit 3D cost volume representation integrated from multiple views. Given a set of depth hypotheses spanning the scene depth range, pixel matching scores are computed over the epipolar lines provided by pose information. Then, 3D convolutional layers are applied as regularization [58] to output a depth map [24, 43, 57] aligned with the RGB view. Volumetric-based methods reconstruct the global scene structure at once back-projecting rays of deep features in a global voxel grid and refining with 3D recurrent layers to finally extract the mesh structure of the scene [4, 10, 42, 47, 48].

DoD differs from MVS methods not only by architectural design but also in that it integrates both multi-view and depth cues. Multi-view and sparse depth fusion have been scarcely investigated in the literature. [39] injects sparse depth in multi-view stereo networks by modulating their 3D cost volume, following [40]. However, such methods struggle to deal with scenes that are not object-centric and usually require a large number of views.

3 Proposed Framework

Our approach consists of exploiting the higher framerate of an RGB camera to increase the temporal resolution of an active depth sensor. This is carried out by leveraging multi-view geometry on the RGB video stream and estimating depth for any RGB frame, both those for which measurements from the active sensor are available and those for which are not. To this aim, we make use of the minimum amount of information needed to exploit geometry – *i.e.* for each RGB view on which we seek to compute depth (the *target* view) we retain a previously collected RGB frame (*source* view) and sparse depth points (*source* depth). The supposed setup is illustrated in Figure 2. We conceptually divide our framework into a set of three sequential steps: i) multi-modal encoding, ii) iterative multi-modal integration, and iii) depth decoding. Layer-wise details of the proposed deep modules are provided in the supplementary material.

3.1 Multi-Modal Encoding

Our framework exploits information from different *modalities* to perceive 3D structures – *i.e.*, multi-view geometry, monocular cues, and sparse depth measurements. To this extent, we define our framework as multi-modal. Accordingly, it is important to properly extract useful cues for each of such information sources. Instead of performing early fusion [19], we separately compute domain-appropriate features and delegate a fusion module, detailed in Section 3.2, to properly integrate them in a common representation. In this section, we specify the encoding of each domain, while Figure 3 depicts an overall overview of DoD.

Geometry Encoding. Multi-view geometry cues stem from the capability to perform matching. We employ the first layers of a ResNet18 [25] to design a shared encoder, used to extract features $\mathcal{F}^t, \mathcal{F}^s$ at $\frac{1}{8}$ spatial resolution, from target and source views respectively. Such features are exploited to compute correlation scores between pixels of the target view and those of the source view. Given the predicted depth at a specific coordinate of the target view D_{u_t, v_t} , the matching coordinates of the same point in the source view can be obtained as

$$q^s = KPD_{u_t, v_t}K^{-1}q^t \quad (1)$$

where K and P are the camera intrinsic parameters and the relative pose between target and source views, while $q^s = [u_s \ v_s \ 1]^T$ and $q^t = [u_t \ v_t \ 1]^T$ are homogeneous point coordinates in the two frames respectively. These latter coordinates allow for sampling features from \mathcal{F}^s and \mathcal{F}^t and compute per-point correlation cues as

$$\mathcal{C} = \frac{1}{\sqrt{F}} \sum_{f=1}^F \mathcal{F}_{u_t v_t f}^t \mathcal{F}_{u_s v_s f}^s \quad (2)$$

However, single pixel-wise correlation scores are not sufficient to guide the depth update process effectively. According to multi-view geometry, as the depth

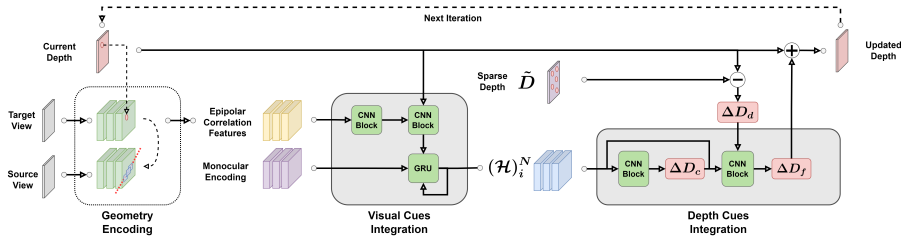


Fig. 3: Depth on Demand Framework Overview. We provide a high-level overview of DoD, level-wise architectural details are provided in the supplementary material. DoD embeds multi-view cues and monocular features in the Visual Cues Integration, then integrates sparse depth updates in the Depth Cues Integration. To properly exploit both these information these stages are applied iteratively in the form of depth updates.

value of a pixel in the target image changes, its corresponding matching pixel in the source view is supposed to move along the epipolar line, moving away or approaching the epipole. Meaningful multi-view cues are guaranteed only if the correct updating direction for estimated depth can be inferred. Thus, for each $D_{u_t v_t}$ we sample a set of depth hypotheses relative to the former, moving along the epipolar line. Then, for each of them, we compute a patch of correlation values to increase the distinctiveness of each sampling. Such procedure generates the “Epipolar Correlation Features” represented in the early stages of Figure 3. If not otherwise specified, we linearly sample 41 3×3 patches within a range of 2 meters, which corresponds to sampling at intervals of 0.1 meters.

Monocular Encoding. Sparse depth information and multi-view correlation data are crucial to deliver accurate 3D reconstructions. Nonetheless, such information fails in the case of moving objects or is not available when large camera pose changes happen, potentially leaving large areas of the field of view empty of information. Thus, it is important to provide a fallback monocular source of information to smoothly complete the not-covered areas. Purposely, we introduce a monocular encoder exploiting the first layers of a ResNet34 [25] to output multi-scale feature maps $\tilde{\mathcal{F}}_2^t$, $\tilde{\mathcal{F}}_4^t$ and $\tilde{\mathcal{F}}_8^t$ at respectively $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$ resolution out of the target view alone.

Sparse Depth Encoding. Finally, we assume the availability of sparse depth data obtained from an active sensor captured at a previous time instant. We project such sparse depth points onto the target view by means of pose information, obtaining a coarse depth map \tilde{D} that will be exploited for both initialization and iterative multi-modal fusion. Since projecting at a lower resolution may lead to inaccurate positioning when building the sparse depth map, we propagate sub-pixel projection coordinates too. Unlike the depth completion task, projected sparse depth is characterized by errors on moving objects and occlusion, making it more difficult to exploit as a source. We delegate their management to the integration phase, where the exploitation of multiple modalities ameliorates such issues.

3.2 Multi-Modal Integration

Once features have been extracted for each input modality, we employ a fusion module to combine such information in the common representation of a target-aligned depth map, that is iteratively refined for a fixed number of steps N . Our integration module is depicted in Figure 3 and can be logically divided into two sequential components.

Visual Cues Integration. The first stage of our fusing module is in charge of extracting depth-related features by visual cues only. Features extracted in the Monocular Encoding step are integrated with the geometric information. This latter consists of a set of correlation features extracted by sampling over the epipolar lines in a relative range with respect to the current depth estimate, as detailed in Section 3.1. We embed all these cues in the hidden state $(\mathcal{H})_i^N$ of a Gated Recurrent Unit, where $(\cdot)_i^N$ indicates a sequence of tensors across a set of iterations from $i = 0$ to $i = N - 1$. We initialize $(\mathcal{H})_{i=0}^N$ with a deep convolutional module fed with monocular features, specified in the supplementary material.

Depth Cues Integration. The second stage of our fusing module takes into account sparse depth data availability. First, a branch predicts a depth update ΔD_c from visual depth-related cues only. Then, a sparse depth update ΔD_d is computed pixel-wise versus the current prediction as

$$\Delta D_d = \begin{cases} \tilde{D}_{i,j} - D_{i,j} & \text{if } \tilde{D}_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

It is worth observing that this latter step generates an update that seeks to refine the current prediction by injecting the exact sparse points, for which zero means either that the depth is correct versus *a priori* information or that the depth measurement is missing for a specific pixel. Finally, the fusion of these updates is carried out by a further branch predicting ΔD_f , which is used to update the current depth prediction. This integration procedure allows for filtering the sparse depth which is likely to contain several outliers due to reprojection – *e.g.*, as in the case of background points being blended with foreground points at occlusions [12]. Moreover, since the sparse depth data is fused in the update space, missing values can be integrated as zero updates, effectively dealing with the varying sparsity problem often affecting depth completion methods [13].

Iterations and Depth Initialization. The previously described multi-modal updating strategy is applied multiple times, generating at each iteration a refined depth map that is then used at the subsequent iteration to improve the multi-view correlation samples and the sparse depth update. Accordingly, an initial depth state is required: we choose to initialize the depth for the first iteration with the sparse depth data, filling the missing coordinates with the mean value of the valid ones. In case no projected sparse depth points are available in the target view we initialize with a reasonable depth value of 3 meters, if not otherwise specified.

3.3 Depth Decoding

The multi-modal integration module outputs a sequence of incrementally refined depth maps $(D)_i^N$ at $\frac{1}{8}$ resolution. While working at a lower resolution is beneficial in terms of memory and computational time, a method to perform effective upsampling is required. We exploit a learned procedure inspired by convex upsampling [50], given the depth map at $\frac{1}{8}$ resolution, we employ a set of three modules $\theta_s(\cdot)$ $s \in \{2, 4, 8\}$ performing a $2\times$ resolution upsampling composed of two convolutional layers. Each module takes in input the depth map to be upsampled, monocular context information $\tilde{\mathcal{F}}_s$ $s \in \{2, 4, 8\}$ and a set of features from the previous step, then it outputs $\frac{H}{s} \times \frac{W}{s} \times M$ feature channels and an upsampling mask \mathcal{W}_s of shape $\frac{H}{s} \times \frac{W}{s} \times (2 \times 2 \times 9)$. This latter is used to perform a weighted combination over the 3×3 neighborhood of each depth value normalized by a softmax operation and yields a $2\times$ upsampled depth map. Features are upsampled by nearest neighbor interpolation. The first module $\theta_8(\cdot)$ takes in input the last hidden state $(\mathcal{H})_{i=N-1}^N$. This approach enables both embedding fine-grain monocular contextual information and enforcing locally smooth and consistent depth propagation. Moreover, it allows for a drastic reduction of the upsampling module weights number with respect to conventional convex upsampling. While optimizing, the upsampling module is applied at each iteration for supervision; however, at deploying time it can be used only once for the final prediction, to maximize efficiency.

4 Experiments

We evaluate our framework in a wide range of scenarios – *i.e.* indoor video sequences, aerial scenes, automotive environments – to evaluate its accuracy within single domains, as well as its generalization capabilities. Since each setting manifests its own challenges, we divide the experiments by scenario and highlight the main difficulties faced on a per-dataset basis.

Training Protocol. To each target frame I_i we associate a buffer of previous N frames $\{(I_j, \tilde{D}_j) : j \in [i - N, i - 1]\}$. Then, at each iteration, we randomly select a frame from such buffer as the source one. This is done to augment as much as possible the number of relative poses observed between the source and target view. We apply random color jitter and horizontal flips, adjusting the pose accordingly. For each sample, we collect a sequence of progressively refined depth maps yielded by the multi-modal integration unit and upsample them to full resolution with the depth decoding approach described in Section 3.3. We supervise such a sequence of depth maps $(D)_i^N$ using an exponentially decayed ℓ_1 -loss, as described in Equation 4 with decaying factor $\nu = 0.8$.

$$L = \sum_{i=1}^N \nu^{N-i} \|(D)_i^N - D_{\text{gt}}\|_1 \quad (4)$$

Testing Protocol. While testing, we suppose an RGB video stream with a higher frame rate than the active depth sensor. Thus, given a sequence of RGB

Table 1: Results on ScanNetV2. On top, (a) 2D and (b) 3D performance by DoD and competing approaches. At the bottom, (c) 3D performance by DoD with low/high temporal resolution. The **best**, **second**-best and **third**-best are highlighted.

Method	Views	2D Metrics					3D Metrics						
		MAE↓	RMSE↓	Abs Rel↓	Sq Rel↓	$\sigma < 1.05$ ↑	Comp↓	Acc↓	Chamfer↓	Prec↑	Recall↑	F-Score↑	
MVS	SimpleRecon [43]	8	0.093	0.151	0.047	0.016	0.717	0.062	0.056	0.059	0.702	0.646	0.671
	PatchMatch-Net [52]	8	0.184	0.270	0.102	0.048	0.437	0.106	0.086	0.096	0.511	0.433	0.467
	CAS-MVSNNet [24]	8	0.170	0.254	0.091	0.044	0.507	0.086	0.082	0.084	0.545	0.498	0.519
	UCS-Net [7]	8	0.167	0.252	0.088	0.042	0.512	0.084	0.082	0.083	0.547	0.502	0.522
MVS + Depth	Guided PatchMatch-Net [52] + [39]	8	0.183	0.267	0.102	0.048	0.437	0.106	0.085	0.095	0.512	0.432	0.467
	Guided CAS-MVSNNet [24] + [39]	8	0.124	0.203	0.068	0.029	0.635	0.064	0.061	0.062	0.667	0.634	0.649
	Guided UCS-Net [7] + [39]	8	0.133	0.210	0.074	0.030	0.576	0.070	0.065	0.068	0.616	0.578	0.595
	Guided PatchMatch-Net [52] + [39]	2	0.291	0.384	0.160	0.096	0.284	0.135	0.125	0.130	0.406	0.315	0.353
Depth	Guided CAS-MVSNNet [24] + [39]	2	0.286	0.388	0.154	0.094	0.304	0.099	0.109	0.104	0.447	0.419	0.431
	Guided UCS-Net [7] + [39]	2	0.258	0.353	0.148	0.093	0.328	0.103	0.099	0.101	0.451	0.385	0.414
	SpAgNet [13]	1	0.069	0.138	0.039	0.016	0.824	0.046	0.037	0.042	0.836	0.789	0.810
	NLSPN [38]	1	0.067	0.137	0.037	0.017	0.847	0.046	0.035	0.041	0.851	0.799	0.822
CompletionFormer [59]	1	0.075	0.149	0.041	0.019	0.829	0.047	0.037	0.042	0.846	0.795	0.818	
DoD (ours)	2	0.041	0.103	0.022	0.008	0.899	0.039	0.025	0.032	0.904	0.845	0.871	

(a)
(b)

Method	3D Metrics					
	Comp↓	Acc↓	Chamfer↓	Prec↑	Recall↑	F-Score↑
DoD - Low Temporal Resolution	0.064	0.014	0.039	0.961	0.778	0.856
DoD - High Temporal Resolution	0.039	0.025	0.032	0.904	0.845	0.871

(c)

frames $[I_0, \dots, I_n]$ only a few of them will be associated with a depth frame $\{\tilde{D}_0, \dots, \tilde{D}_m\}$. In each testing video sequence, we link to each RGB view I_i the immediately preceding RGB image coupled with a depth frame (I_j, \tilde{D}_j) , $j \leq i$ and feed our and competing methods with such data to predict a dense depth map. By modulating the temporal sparsification ratio between the depth and RGB frames $\tau = f_D/f_{\text{RGB}}$ we can control how frequently the sparse depth frames are provided; with ratio 1 the task is equivalent to depth completion as depth would be given at every frame.

Competitors. To fairly compare with existing approaches, we retrain each one following the authors’ guidelines but applying the previously described training protocol to increase their robustness to the peculiarities of the proposed task since the standard original training protocol for depth completion struggles at dealing with the considered setup. Concerning depth completion, we compare with state-of-the-art frameworks [12, 38, 59] projecting sparse depth points from the source frame onto the target view I_i . Concerning Multi-View Stereo methods, we select [39] as a natural competitor since it enables standard MVS frameworks [7, 24, 52] to exploit both sparse depth and multi-view data natively. Also in this latter case, we train with the aforementioned scheme. Since Multi-View Stereo methods usually exploit a large number of views, we train and evaluate with both 2 and 8 input views.

4.1 Indoor Scenario

For indoor applications, ToF sensors are a popular solution to perceive depth, even though they are characterized by relatively low spatial resolution and a short working range. In this setting, we limit the temporal resolution of the ToF sensor as well, in order to reduce power consumption and overheating. The main challenge in indoor environments is the FoV overlap across consecutive views, which may vary from being almost complete – *e.g.*, the camera is not moving – to

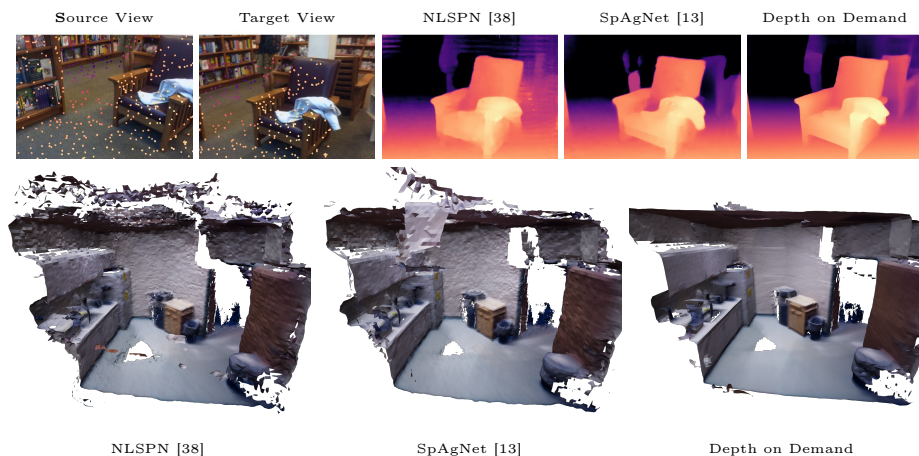


Fig. 4: Qualitative results on ScanNetV2. On top: from left to right the source view with sparse depth points, the target view with projected sparse depth points, and predictions by competitors and DoD. At the bottom: reconstructed meshes by competitors and DoD, respectively at low and high temporal resolution.

completely absent. We train on ScanNetV2 [14] and test on both this latter and 7Scenes [22], following the protocols described in Section 4 by randomly sampling 500 sparse depth points consistently with the depth completion literature [8, 32, 38]. For testing, we sparsify depth over time according to $\tau = 0.2$.

ScanNetV2. ScanNetV2 [14] is an RGB-D video dataset containing more than 1500 scans of indoor environments. Table 1 (a) shows the 2D performance of our framework on standard metrics for depth map evaluation. At the top, we report the performance of RGB-only methods [7, 24, 43, 52] with 8 views in input. Below, we show the performance of [39] with either 8 or 2 input views and projected sparse depth. In such methods, integrating sparse depth in our scenario provides a small improvement, nullified by using only 2 views, *i.e.* the target and a single source view. We ascribe this to their specific design, poor at processing sequential frames. On the contrary, depth completion methods [13, 38, 59] relying only on the target view and projected sparse depth from the source view – showed at the bottom of Table 1 – result in being the most competitive solution for temporal depth stream densification among those existing in the literature. Eventually, our framework indisputably outperforms completion models on any metric, thanks to the joint use of monocular, multi-view, and sparse depth cues. This superior accuracy of the predicted depth maps also translates into more accurate, dense 3D reconstructions: Table 1 (b) shows how even a few temporally sparse depth measurements largely improve performance versus RGB-only reconstruction carried out by state-of-the-art methods. Again, our framework outperforms existing depth completion solutions by an evident margin. Table 1 (c), instead, highlights the effect of increasing the temporal resolution at which depth is estimated. We can notice how keeping a low temporal resolution – *i.e.*,

Table 2: Results on 7Scenes. 2D performance by DoD and competing approaches in generalization on 7Scenes. The **best**, **second -best** and **third -best** are highlighted.

	Method	Views	2D Metrics				
			MAE↓	RMSE↓	Abs Rel↓	Sq Rel↓	$\sigma < 1.05↑$
MVS	SimpleRecon [43]	8	0.121	0.169	0.068	0.021	0.536
	PatchMatch-Net [52]	8	0.193	0.268	0.112	0.048	0.390
	CAS-MVSNet [24]	8	0.177	0.251	0.101	0.041	0.421
	UCS-Net [7]	8	0.176	0.250	0.099	0.040	0.428
MVS + Depth	Guided PatchMatch-Net [52]+ [39]	8	0.191	0.264	0.112	0.047	0.391
	Guided CAS-MVSNet [24]+ [39]	8	0.120	0.192	0.071	0.024	0.587
	Guided UCS-Net [7]+ [39]	8	0.141	0.209	0.083	0.028	0.484
MVS + Depth	Guided PatchMatch-Net [52]+ [39]	2	0.267	0.345	0.158	0.080	0.268
	Guided CAS-MVSNet [24]+ [39]	2	0.250	0.338	0.141	0.069	0.303
	Guided UCS-Net [7]+ [39]	2	0.228	0.306	0.139	0.063	0.303
Depth	SpAgNet [13]	1	0.068	0.139	0.040	0.014	0.806
	NLSPN [38]	1	0.061	0.134	0.037	0.014	0.842
	CompletionFormer [59]	1	0.067	0.144	0.039	0.015	0.827
	DoD (ours)	2	0.043	0.106	0.025	0.008	0.896

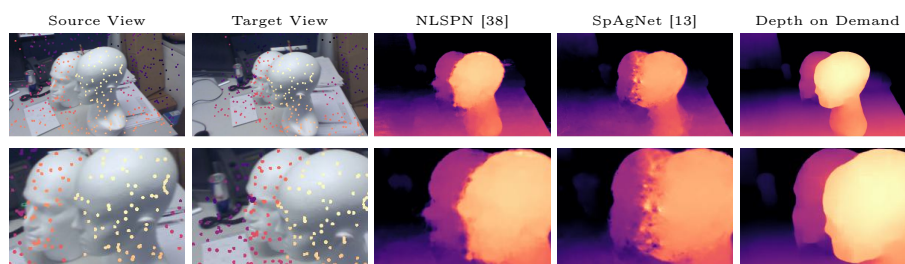


Fig. 5: Qualitative results on 7Scenes. From left to right: source view with sparse depth points, the target view with projected sparse depth points, and predictions by DoD and existing methods.

the same as the depth sensors – yields slightly accurate reconstructed meshes, while a higher temporal resolution trades accuracy to increase completeness. Nonetheless, maintaining a high temporal resolution yields better F-Scores overall. Finally, Figure 4 shows some qualitative results.

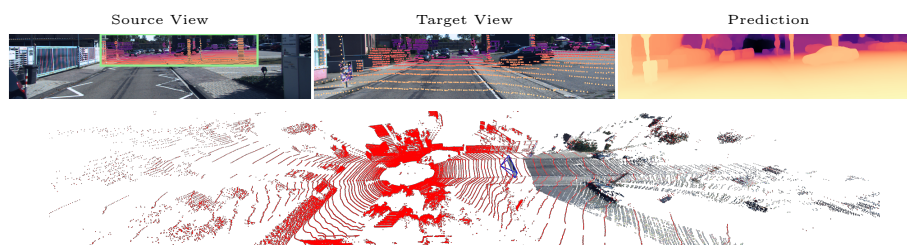
7Scenes. We assess the generalization capabilities of our method and the existing alternatives in different indoor environments on the 7Scenes dataset [46], by testing the models trained on ScanNetV2 [14] without any fine-tuning. Results are collected in Table 2, where we can notice a trend consistent with what was observed on ScanNetV2: our framework shows remarkable capabilities concerning generalization in the indoor scenario, staying in the lead of the competing approaches. In Figure 5 we provide a comparison of handling erroneous sparse depth points due to occlusion where DoD is able to disregard outliers by exploiting multi-view cues.

4.2 Outdoor Scenario

3D reconstruction in outdoor environments poses significantly different challenges compared to indoor – *e.g.*, it features much larger depth ranges and, possibly, scattering of the depth measurements. To study temporal depth completion in this context, we exploit two datasets: TartanAir [53] and KITTI [21].

Table 3: Results on TartanAir. 2D performance of our and competing approaches on TartanAir [53]. The **best**, **second**-best and **third**-best are highlighted.

	Method	Views	2D Metrics				
			MAE↓	RMSE↓	Abs Rel↓	Sq Rel↓	$\sigma < 1.05$ ↑
MVS + Depth	Guided PatchMatch-Net [52]+ [39]	8	2.353	5.259	0.234	2.285	0.470
	Guided CAS-MVSNet [24]+ [39]	8	1.296	3.753	0.126	1.270	0.647
	Guided UCS-Net [7]+ [39]	8	1.231	3.624	0.115	1.106	0.675
MVS + Depth	Guided PatchMatch-Net [52]+ [39]	2	3.629	6.564	0.438	3.669	0.230
	Guided CAS-MVSNet [24]+ [39]	2	1.985	4.845	0.185	1.794	0.492
	Guided UCS-Net [7]+ [39]	2	1.804	4.513	0.177	1.526	0.486
Depth	SpAgNet [13]	1	0.841	2.273	0.090	0.561	0.718
	NLSPN [38]	1	0.941	2.327	0.113	0.623	0.613
	CompletionFormer [59]	1	0.961	2.411	0.106	0.608	0.625
	DoD (ours)	2	0.648	2.230	0.056	0.490	0.832

**Fig. 6: KITTI Setup.** On KITTI, we project the 360° LiDAR point cloud over the target point of view. If the camera is moving forward – as usually happens – the furthest scan lines are used only, leading to noisy and spaced depth values on the target view. However, the FoV of the target image is usually fully covered.

TartanAir. TartanAir [53] is a large synthetic dataset featuring photo-realistic environments with different weather and light conditions. It provides a drone-like point of view in a wide set of scenarios featuring high-frequency details and fast camera motion. Table 3 collects the results achieved by existing methods combining multi-view geometry and sparse depth measurements [39] or performing depth completion [13, 38, 59] and our framework. As for the indoor case, completion models largely outperform competitor networks, confirming the limitations of these latter at dealing with the considered problem. Again, our architecture shines in accuracy, achieving the lowest errors by a notable margin.

KITTI. The KITTI [21] dataset is a well-known outdoor benchmark with LiDAR data, widely used for visual odometry, monocular depth prediction, and depth completion. For automotive applications, 360° LiDAR sensors are usually employed, providing long-range depth at a frequency limited by the revolution time required by the rotating laser beams. Thus, despite the color cameras can acquire frames at a much higher rate, this is usually constrained to the LiDAR frame rate when performing tasks exploiting both – *e.g.*, depth completion. Nonetheless, when the LiDAR scans are projected from a previous frame over a consequent one as we do to perform temporal depth completion, the target FoV is almost always fully covered with sparse depth points, yet with higher spatial sparsity. Figure 6 shows an example where a LiDAR point cloud collected at a

Table 4: Results on KITTI. 2D performance by DoD and competing approaches. The **best**, **second**-best and **third**-best are highlighted.

Method	Views	2D Metrics – LiDAR 1Hz					2D Metrics – LiDAR 0.5Hz					
		MAE↓	RMSE↓	Abs Rel↓	Sq Rel↓	$\sigma < 1.05↑$	MAE↓	RMSE↓	Abs Rel↓	Sq Rel↓	$\sigma < 1.05↑$	
10Hz + Depth	Guided PatchMatch-Net [52]+ [39]	8	2.649	5.149	0.216	3.090	0.453	2.809	5.353	0.232	3.330	0.416
	Guided CAS-MVSNNet [24]+ [39]	8	0.608	2.126	0.034	0.229	0.888	0.873	2.501	0.052	0.350	0.786
	Guided UCS-Net [7]+ [39]	8	0.575	1.930	0.034	0.229	0.881	0.828	2.321	0.050	0.303	0.789
10Hz + Depth	Guided PatchMatch-Net [52]+ [39]	2	1.898	4.165	0.117	0.863	0.496	2.282	4.564	0.145	1.154	0.404
	Guided CAS-MVSNNet [24]+ [39]	2	0.676	2.203	0.035	0.225	0.872	0.916	2.562	0.052	0.328	0.773
	Guided UCS-Net [7]+ [39]	2	0.545	1.859	0.030	0.146	0.885	0.837	2.330	0.049	0.277	0.779
Depth	SpAgNet [13]	1	0.532	1.626	0.027	0.095	0.879	0.687	1.865	0.037	0.133	0.808
	NLSFN [38]	1	0.426	1.282	0.023	0.069	0.902	0.614	1.591	0.035	0.121	0.827
	CompletionFormer [59]	1	0.348	1.299	0.019	0.085	0.939	0.555	1.695	0.031	0.150	0.868
	DoD (ours)	2	0.347	1.288	0.017	0.061	0.944	0.492	1.544	0.025	0.094	0.890

certain time frame is projected over an RGB image collected thereafter, with the camera having moved forward in between the two acquisitions. This causes only the furthest scan lines to be projected over the target view, looking sparser, noisier, and manifesting errors due to occlusions or moving objects. Large areas missing any depth measure may occur in case of occlusion caused by objects in the source view, but still, the FoV is usually fully covered. In this scenario, our approach is at a disadvantage compared to other methods cause i) the reduced multi-view visual overlap on long distances does not provide large benefits and ii) the spatial distribution of the depth measurements is more steady.

Table 4 shows the results achieved by existing methods and ours on this dataset, by simulating different temporal sparsification levels – *i.e.* RGB camera at 10Hz and the LiDAR sensor at respectively 1Hz and 0.5Hz. We exploit an off-the-shelf keypoint matcher [33], perspective-n-points [30], and locally-optimized RANSAC [11] to estimate accurate pose, as already done in the depth completion literature [37]. Despite the more challenging setting, our framework still outperforms any existing alternative.

4.3 Temporal Sparsification Study

We study the sensitivity of our approach to different *temporal* densities, *i.e.*, frame rate imbalances between the RGB and depth sensor (that is, $\tau = f_D/f_{RGB}$). In Figure 7(a) we report the Mean Absolute Error (MAE) on the 7Scenes test split with the testing protocol described in Section 4, while varying the temporal sparsification τ from 0.1 – *i.e.* one out of ten frames – to 1. Actually, when $\tau = 1$ the source view always matches with the target view, and sparse points are aligned with it. Thus, $\tau = 1$ is equivalent to the well-studied depth completion case. This may also occur in real use cases where the camera is static. We report a sensitivity study to different *spatial* densities in the supplementary material.

4.4 Memory and Time Analysis

Figures 7 (b) and 7 (c) report the memory footprint and execution time of the main methods involved in our experiments. We measure the peak memory and computation time the model requires for inference when processing 640×480 inputs. All measurements are based on a single RTX 3090 GPU and 32-bit floating-point precision. Our approach excels in terms of both.

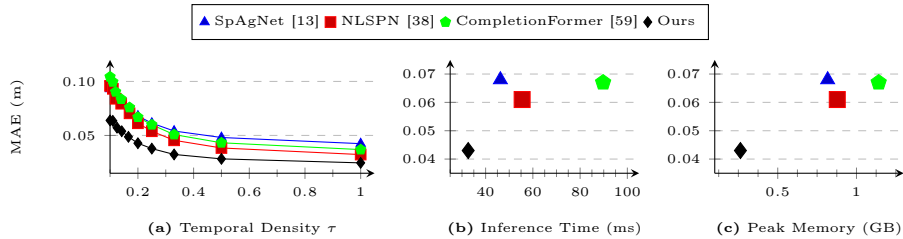


Fig. 7: Memory and Time Study. We analyze time and memory footprint in evaluation on a single RTX 3090 GPU of our and competing methods 7Scenes.

Table 5: Ablation studies. Experiments on ScanNetV2 aimed at highlighting (a) the impact of multi-view stereo and depth cues, and (b) runtime for each component.

MVS	Depth	2D Metrics		3D Metrics	
		MAE \downarrow	RMSE \downarrow	Chamfer \downarrow	F-Score \uparrow
		0.187	0.257	0.084	0.530
✓		0.049	0.115	0.033	0.870
✓	✓	0.041	0.103	0.032	0.871

(a)

Module	Single Inf. Time	Calls	Tot. Time
	(ms)	(nr.)	(ms)
Geometry Encoding	2.060 \pm 0.182	1 \times	2.060 \pm 0.182
Monocular Encoding	2.175 \pm 0.031	1 \times	2.175 \pm 0.031
Correlation Features	0.373 \pm 0.001	10 \times	3.733 \pm 0.015
Visual Cues Integration	1.748 \pm 0.008	10 \times	17.480 \pm 0.082
Depth Cues Integration	0.274 \pm 0.005	10 \times	2.724 \pm 0.053
Depth Decoding	1.444 \pm 0.040	1 \times	1.444 \pm 0.040
Total Time			30.196 \pm 0.260

(b)

4.5 Ablation study

Finally, we conclude with an ablation study to assess the impact of each module composing DoD. Table 5 reports results on ScannetV2 concerning two main studies. In (a), we show how processing multi-view stereo cues and sparse depth impacts the overall accuracy achieved by DoD. Not surprisingly, depth points play a prevalent role, yet alone are insufficient to achieve the best results. In (b), we report the detailed runtime required by any single component in DoD.

5 Conclusion

In this paper, we faced the temporal sparsification of a video RGB-D stream when reducing *temporally* the number of depth frames used for accurate 3D reconstruction. This peculiar setup aims to decrease active depth sensors’ energy consumption or overcome their limited frame rate compared to cameras. Purposely, we proposed an approach to integrate depth, monocular, and multi-view cues in a remarkably effective common framework, as confirmed by the extensive validation over various datasets. Additionally, our proposal features a significantly lower memory footprint and execution time than competitors.

Limitations. The presence of moving objects inherently harms DoD accuracy, yet without catastrophic failures (see the supplementary material). Future work will focus on this direction to further improve DoD in these occurrences.

Acknowledgment. We gratefully acknowledge Sony Depthsensing Solutions SA/NV for funding this research.

References

1. Bamji, C., Godbaz, J., Oh, M., Mehta, S., Payne, A., Ortiz, S., Nagaraja, S., Perry, T., Thompson, B.: A review of indirect time-of-flight technologies. *IEEE Transactions on Electron Devices* **69**(6), 2779–2793 (2022). <https://doi.org/10.1109/TED.2022.3145762>
2. Bartoccioni, F., Zablocki, É., Pérez, P., Cord, M., Alahari, K.: Lidartouch: Monocular metric depth estimation with a few-beam lidar. *Computer Vision and Image Understanding* **227**, 103601 (2023)
3. Bhandari, A., Raskar, R.: Signal processing for time-of-flight imaging sensors: An introduction to inverse problems in computational 3-d imaging. *IEEE Signal Processing Magazine* **33**(5), 45–58 (2016). <https://doi.org/10.1109/MSP.2016.2582218>
4. Bozic, A., Palafox, P., Thies, J., Dai, A., Nießner, M.: TransformerFusion: Monocular RGB scene reconstruction using transformers. *NeurIPS* (2021)
5. Campbell, N.D., Vogiatzis, G., Hernández, C., Cipolla, R.: Using multiple hypotheses to improve depth-maps for multi-view stereo. In: *European Conference on Computer Vision*. pp. 766–779. Springer (2008)
6. Chen, Y., Ren, J.S.J., Cheng, X., Qian, K., Gu, J.: Very power efficient neural time-of-flight. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)* pp. 2246–2255 (2018), <https://api.semanticscholar.org/CorpusID:56475963>
7. Cheng, S., Xu, Z., Zhu, S., Li, Z., Li, L.E., Ramamoorthi, R., Su, H.: Deep stereo using adaptive thin volume representation with uncertainty awareness. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 2524–2534 (2020)
8. Cheng, X., Wang, P., Yang, R.: Depth estimation via affinity learned with convolutional spatial propagation network. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. pp. 103–119 (2018)
9. Cheng, X., Wang, P., Yang, R.: Learning depth with convolutional spatial propagation network. *IEEE transactions on pattern analysis and machine intelligence* (2019)
10. Choe, J., Im, S., Rameau, F., Kang, M., Kweon, I.S.: VolumeFusion: Deep depth fusion for 3D scene reconstruction. In: *ICCV* (2021)
11. Chum, O., Matas, J., Kittler, J.: Locally optimized ransac. In: *DAGM-Symposium* (2003), <https://api.semanticscholar.org/CorpusID:15181392>
12. Conti, A., Poggi, M., Aleotti, F., Mattoccia, S.: Unsupervised confidence for lidar depth maps and applications. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2022), iROS
13. Conti, A., Poggi, M., Mattoccia, S.: Sparsity agnostic depth completion. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. pp. 5871–5880 (January 2023)
14. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: *Proc. Computer Vision and Pattern Recognition (CVPR)*, IEEE (2017)
15. Dimitrievski, M.D., Veelaert, P., Philips, W.: Learning morphological operators for depth completion. In: *Advanced Concepts for Intelligent Vision Systems Conference* (2018)
16. Eldesokey, A., Felsberg, M., Khan, F.S.: Propagating confidences through cnns for sparse data regression. In: *British Machine Vision Conference* (2018), <https://api.semanticscholar.org/CorpusID:44081968>

17. Fan, R., Li, Z., Poggi, M., Mattoccia, S., et al.: A cascade dense connection fusion network for depth completion. (2022)
18. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence* **32**(8), 1362–1376 (2009)
19. Gadzicki, K., Khamsehashari, R., Zetzsche, C.: Early vs late fusion in multimodal convolutional neural networks. In: 2020 IEEE 23rd international conference on information fusion (FUSION). pp. 1–6. IEEE (2020)
20. Galliani, S., Lasinger, K., Schindler, K.: Massively parallel multiview stereopsis by surface normal diffusion. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 873–881 (2015)
21. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
22. Glocker, B., Izadi, S., Shotton, J., Criminisi, A.: Real-time rgb-d camera relocation. In: International Symposium on Mixed and Augmented Reality (ISMAR). IEEE (October 2013)
23. Gu, J., Xiang, Z., Ye, Y., Wang, L.: Denselidar: A real-time pseudo dense depth guided depth completion network. *IEEE Robotics and Automation Letters* **6**, 1808–1815 (2021)
24. Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2495–2504 (2020)
25. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 770–778 (2015), <https://api.semanticscholar.org/CorpusID:206594692>
26. Hu, J., Bao, C., Ozay, M., Fan, C., Gao, Q., Liu, H., Lam, T.L.: Deep depth completion from extremely sparse data: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* pp. 1–20 (2022). <https://doi.org/10.1109/TPAMI.2022.3229090>
27. Hu, M., Wang, S., Li, B., Ning, S., Fan, L., Gong, X.: Towards precise and efficient image guided depth completion (2021)
28. Imran, S.M., Long, Y., Liu, X., Morris, D.: Depth coefficients for depth completion. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 12438–12447 (2019)
29. Jiang, X., Cambareri, V., Agresti, G., Ugwu, C.I., Simonetto, A., Cardinaux, F., Zanuttigh, P.: A low memory footprint quantized neural network for depth completion of very sparse time-of-flight depth maps. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 2687–2696 (June 2022)
30. Lepetit, V., Moreno-Noguer, F., Fua, P.: Epanp: An accurate $o(n)$ solution to the pnp problem. *International Journal of Computer Vision* **81**, 155–166 (2009)
31. Li, A., Yuan, Z., Ling, Y., Chi, W., Zhang, S., Zhang, C.: A multi-scale guided cascade hourglass network for depth completion. 2020 IEEE Winter Conference on Applications of Computer Vision (WACV) pp. 32–40 (2020)
32. Lin, Y., Cheng, T., Zhong, Q., Zhou, W., Yang, H.: Dynamic spatial propagation network for depth completion. vol. 36 (2022). <https://doi.org/10.1609/aaai.v36i2.20055>
33. Lindenberger, P., Sarlin, P.E., Pollefeys, M.: LightGlue: Local Feature Matching at Light Speed. In: ICCV (2023)

34. Lopez-Rodriguez, A., Busam, B., Mikolajczyk, K.: Project to adapt: Domain adaptation for depth completion from noisy and sparse sensor data. In: Proceedings of the Asian Conference on Computer Vision (2020)
35. Lu, K., Barnes, N., Anwar, S., Zheng, L.: From depth what can you see? depth completion via auxiliary image reconstruction. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 11303–11312 (2020), <https://api.semanticscholar.org/CorpusID:219615769>
36. Luetzenburg, G., Kroon, A., Bjørk, A.A.: Evaluation of the Apple iPhone 12 Pro LiDAR for an Application in Geosciences. *Scientific Reports* **11**(1) (Nov 2021). <https://doi.org/10.1038/s41598-021-01763-9>, <https://doi.org/10.1038/s41598-021-01763-9>
37. Ma, F., Cavalheiro, G.V., Karaman, S.: Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. 2019 International Conference on Robotics and Automation (ICRA) pp. 3288–3295 (2018)
38. Park, J., Joo, K., Hu, Z., Liu, C.K., Kweon, I.S.: Non-local spatial propagation network for depth completion. In: Proc. of European Conference on Computer Vision (ECCV) (2020)
39. Poggi, M., Conti, A., Mattocchia, S.: Multi-view guided multi-view stereo. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2022), iROS
40. Poggi, M., Pallotti, D., Tosi, F., Mattocchia, S.: Guided stereo matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
41. Qiao, X., Poggi, M., Deng, P., Wei, H., Ge, C., Mattocchia, S.: Rgb guided tof imaging system: A survey of deep learning-based methods. *Int. J. Comput. Vis.* (2024), <https://link.springer.com/article/10.1007/s11263-024-02089-5>
42. Rich, A., Stier, N., Sen, P., Höllerer, T.: 3dvnnet: Multi-view depth prediction and volumetric refinement. In: International Conference on 3D Vision (3DV) (2021)
43. Sayed, M., Gibson, J., Watson, J., Prisacariu, V., Firman, M., Godard, C.: Simplecon: 3d reconstruction without 3d convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022)
44. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision. pp. 501–518. Springer (2016)
45. Senushkin, D., Belikov, I., Konushin, A.: Decoder modulation for indoor depth completion. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) pp. 2181–2188 (2021)
46. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.W.: Scene coordinate regression forests for camera relocalization in rgb-d images. 2013 IEEE Conference on Computer Vision and Pattern Recognition pp. 2930–2937 (2013), <https://api.semanticscholar.org/CorpusID:8632684>
47. Stier, N., Rich, A., Sen, P., Höllerer, T.: Vortex: Volumetric 3d reconstruction with transformers for voxelwise view selection and fusion. In: International Conference on 3D Vision (3DV) (2021)
48. Sun, J., Xie, Y., Chen, L., Zhou, X., Bao, H.: NeuralRecon: Real-time coherent 3D reconstruction from monocular video. In: CVPR (2021)
49. Tang, J., Tian, F.P., An, B., Li, J., Tan, P.: Bilateral propagation network for depth completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9763–9772 (June 2024)

50. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) *Computer Vision – ECCV 2020*. pp. 402–419. Springer International Publishing, Cham (2020)
51. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant cnns. In: *2017 International Conference on 3D Vision (3DV)*. pp. 11–20 (2017). <https://doi.org/10.1109/3DV.2017.00012>
52. Wang, F., Galliani, S., Vogel, C., Speciale, P., Pollefeys, M.: Patchmatchnet: Learned multi-view patchmatch stereo. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 14194–14203 (2021)
53. Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., Scherer, S.: Tartanair: A dataset to push the limits of visual slam (2020)
54. Yan, Z., Lin, Y., Wang, K., Zheng, Y., Wang, Y., Zhang, Z., Li, J., Yang, J.: Tri-perspective view decomposition for geometry-aware depth completion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4874–4884 (2024)
55. Yan, Z., Wang, K., Li, X., Zhang, Z., Li, J., Yang, J.: Rignet: Repetitive image guided network for depth completion (2022)
56. Yan, Z., Wang, K., Li, X., Zhang, Z., Li, J., Yang, J.: Desnet: Decomposed scale-consistent network for unsupervised depth completion. In: *Proceedings of the AAAI conference on artificial intelligence*. vol. 37, pp. 3109–3117 (2023)
57. Yang, J., Mao, W., Alvarez, J.M., Liu, M.: Cost volume pyramid based depth inference for multi-view stereo. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4877–4886 (2020)
58. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. pp. 767–783 (2018)
59. Zhang, Y., Guo, X., Poggi, M., Zhu, Z., Huang, G., Mattoccia, S.: Completion-former: Depth completion with convolutions and vision transformers. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 18527–18536 (June 2023)
60. Zhao, S., Gong, M., Fu, H., Tao, D.: Adaptive context-aware multi-modal network for depth completion. *IEEE Transactions on Image Processing* **30**, 5264–5276 (2020)